

Aplicação de um Sistema Para Ajuste de Parâmetros de Navegação Para Robôs Autônomos Como Ferramenta de Ensino

Alexsandro Cavalcanti Chaves Filho
Grupo de Pesquisas em Computação
Física
CESAR SCHOOL
Recife, Brasil
accf3@cesar.school

Henrique Braga Foresti
Grupo de Pesquisas em Computação
Física
CESAR SCHOOL
Recife, Brasil
hbf@cesar.school

Diocleciano Dantas Neto
Grupo de Pesquisas em Computação
Física
CESAR SCHOOL
Recife, Brasil
ddn@cesar.school

Resumo—O presente artigo demonstra um estudo sobre tecnologias utilizadas em sistemas de telemetria, como a comunicação via Bluetooth utilizando a placa ESP32 da Espressif Systems. É apresentada uma aplicação que realiza a configuração de parâmetros para uma abordagem de controle Proporcional, Integral e Derivativa (PID), implementada em robôs de navegação autônoma. O trabalho também explora as tecnologias em contextos educacionais, aplicando uma formação com jovens de ensino técnico por meio do envio e recebimento de comandos no sistema implementado, além disso, estimula o desenvolvimento de circuitos eletrônicos e softwares de controle com placas microcontroladas. A pesquisa enfatiza a prática dessas tecnologias em sala de aula, oferecendo uma ferramenta eficaz para o ensino de conceitos avançados de robótica e automação.

Abstract— This article demonstrates a study of technologies used in telemetry systems, such as Bluetooth communication using the ESP32 board from Espressif Systems. An application is presented that configures parameters for a Proportional, Integral, and Derivative (PID) control approach, implemented in autonomous navigation robots. The work also explores the technologies in educational contexts, applying training to young technical education students through sending and receiving commands in the implemented system, in addition, it encourages the development of electronic circuits and control software with microcontroller boards. The research emphasizes the practice of these technologies in the classroom, offering an effective tool for teaching advanced concepts of robotics and automation.

Keywords—ESP32, Telemetria, PID, Educação

I. INTRODUÇÃO

A robótica educacional tem se mostrado uma ferramenta eficaz para o desenvolvimento de habilidades cognitivas, socioemocionais e tecnológicas em estudantes. Entre as várias aplicações, os robôs seguidores de linhas são especialmente populares, introduzindo conceitos de mecânica, eletrônica e programação de forma prática e envolvente.

Este artigo investiga a aplicação da telemetria em robôs seguidores de linhas, destacando como a coleta e análise de dados podem otimizar o desempenho desses robôs e enriquecer o aprendizado dos estudantes. A pesquisa se baseia na aplicação prática da tecnologia, buscando um meio de estabelecer a conexão entre o robô e o computador de modo mais eficiente e intuitiva possível, facilitando o uso para equipes que competem em tal modalidade.

Os resultados esperados devem demonstrar a eficácia da telemetria como uma ferramenta educacional, oferecendo insights para a integração dessa tecnologia no ensino da robótica.

A. A placa micro controlada ESP-32

A ESP32 é uma placa de desenvolvimento de hardware baseado em um sistema dual core de 32 bits, amplamente utilizado em aplicações de Internet das Coisas (IoT). Ela é projetada para ser uma solução de baixo custo e baixo consumo de energia, tornando-o ideal para aplicações que requerem longa vida útil e manutenção de baixo custo.

A presença do módulo Bluetooth na ESP32 é uma das principais características que a tornam tão útil em aplicações IoT. O módulo Bluetooth permite que o sistema se comunique com dispositivos móveis, como smartphones e tablets, possibilitando que os usuários controlem e monitorem os dispositivos IoT remotamente.

B. Teoria de controle PID

O controle PID (Proporcional-Integral-Derivativo) é um sistema de controle de malha fechada onde se utiliza a realimentação do sinal de saída para ajustar continuamente o controlador e minimizar o erro, alinhando a saída ao valor desejado (Ogata, 2010)[1]. Ele é amplamente utilizado na indústria e em sistemas de automação devido à sua versatilidade e eficácia. Combinando três componentes – proporcional, integral e derivativo – o PID oferece um controle preciso e eficiente, adaptando-se a uma ampla gama de aplicações. Esta técnica é valorizada por proporcionar uma performance robusta e confiável, sendo a escolha preferida por engenheiros e profissionais de automação (Ogata, 2010)[2].

Esta estratégia de controle combina três ações distintas para obter um desempenho otimizado do sistema:

Ação Proporcional (P): Essa ação é responsável por gerar uma resposta proporcional ao erro entre o valor desejado e o valor medido. Ela fornece uma correção imediata, mas pode levar a erros de regime permanente.

Ação Integral (I): A ação integral acumula o erro ao longo do tempo e gera uma resposta que tende a eliminar o erro de regime permanente. No entanto, ela pode causar instabilidade e oscilações no sistema.

Ação Derivativa (D): A ação derivativa antecipa as mudanças no erro e gera uma resposta proporcional à taxa de variação do erro. Isso ajuda a amortecer as oscilações e melhorar a estabilidade do sistema.

A combinação dessas três ações (P, I e D) em um controlador PID permite obter um desempenho superior em relação a cada uma delas individualmente. O ajuste adequado dos parâmetros do controlador PID (ganhos proporcional, integral e derivativo) é crucial para obter a resposta desejada do sistema, como tempo de resposta, sobressinal e estabilidade.

C. Bluetooth Low Energy (BLE): Uma Evolução Eficiente para a Conectividade Curto Alcance

O Bluetooth Low Energy (BLE), também conhecido como Bluetooth Smart, surgiu como uma alternativa revolucionária ao Bluetooth clássico, otimizando o consumo de energia e expandindo as possibilidades da tecnologia. O BLE se destaca por sua capacidade de operar com baixíssimas correntes, permitindo que dispositivos como sensores, wearables e beacons funcionem por longos períodos com baterias pequenas. Esta característica o torna ideal para aplicações na Internet das Coisas (IoT), onde a autonomia e a longevidade dos dispositivos são cruciais (Bluetooth Core Specification Version 5.2 Feature Overview, 2024)[3].

Ao contrário do Bluetooth clássico, que prioriza a velocidade de transmissão, o BLE foca na eficiência energética, utilizando um protocolo de comunicação mais simples e menos exigente. Isso resulta em menor consumo de energia, permitindo que dispositivos BLE operem por mais tempo com a mesma bateria (Colom-Palacios et al., 2014)[4]. Essa otimização energética abre caminho para uma ampla gama de aplicações, desde monitoramento de saúde e fitness até rastreamento de ativos e automação residencial.

Vantagens do BLE:

Baixo consumo de energia: O BLE consome até 100 vezes menos energia que o Bluetooth clássico, permitindo que dispositivos operem por longos períodos com baterias pequenas.

Alcance ampliado: O BLE pode alcançar distâncias maiores que o Bluetooth clássico, permitindo a comunicação entre dispositivos mesmo em ambientes com obstáculos.

Conectividade multiponto: Um único dispositivo BLE pode se conectar a vários outros ao mesmo tempo, facilitando a criação de redes complexas e interligadas.

Alta confiabilidade: O BLE utiliza técnicas avançadas de modulação e salto de frequência para garantir a confiabilidade da comunicação, mesmo em ambientes com interferências.

II. APLICAÇÃO DO SISTEMA DE TELEMETRIA

O sistema em questão usa a tecnologia BLE para viabilizar a comunicação entre um dispositivo ESP32 e um PC. O ESP32, atuando como periférico, coleta dados e os transmite para o PC via BLE. No computador pessoal, o PySerial, uma biblioteca em Python que é usada para ler portas seriais, é utilizada para viabilizar o recebimento dos dados.

Essa arquitetura oferece diversas vantagens:

- **Baixo consumo de energia:** O BLE é conhecido por sua eficiência energética, tornando-o ideal para aplicações que exigem longa vida útil da bateria, como dispositivos IoT.
- **Comunicação confiável:** O BLE garante uma conexão robusta e confiável entre o ESP32 e o PC, mesmo em ambientes com interferências.
- **Fácil implementação:** O PySerial é uma biblioteca Python amplamente utilizada para comunicação serial, facilitando o desenvolvimento e a integração do sistema no PC.
- **Flexibilidade:** O sistema permite a comunicação de diversos tipos de dados entre o ESP32 e o PC, atendendo a uma ampla gama de necessidades.

A. Detectando a porta de comunicação

Existe um protocolo de comunicação onde o ESP32 entra em um loop infinito, enviando periodicamente mensagens Bluetooth com um identificador único, como ilustrado na figura 1. O computador, por sua vez, efetua uma varredura contínua em busca de dispositivos Bluetooth próximos. Ao detectar a mensagem do ESP32, o computador registra a porta Bluetooth associada à recepção e envia uma mensagem de resposta para o ESP32 através da porta detectada, como ilustrado na figura 2. Essa resposta contém o identificador único do ESP32 e a porta Bluetooth do computador. Ao receber a mensagem de resposta, o ESP32 interrompe o envio de mensagens em loop e armazena a porta Bluetooth do computador em sua memória para uso em comunicações subsequentes.

```
void loop() {
  if (SerialBT.available()) {
    String receivedMsg = SerialBT.readStringUntil('\n'); // Lê a mensagem recebida via Bluetooth até encontrar uma quebra de linha
    receivedMsg.trim(); // Remove espaços em branco do início e do final da mensagem

    // Verifica se a mensagem recebida é a que indica que a porta foi encontrada
    if (receivedMsg == "Porta achada") {
      SerialBT.println("porta de saída");
      correndo(); // Chama a função correndo
    }
  }
}
```

Fig. 1. Código na ESP32 que envia a mensagem e verifica se a porta foi achada

```
def find_port(keyword):
    for port in serial.tools.list_ports.comports():
        try:
            ser = serial.Serial(port.device, baudrate=9600)
            ser.write("Porta achada".encode())
            response = ser.readline().decode().strip()
            if response == "porta de saída":
                ser.write("Porta achada\n".encode()) # Escrevendo a mensagem na porta
                ser.close()
                return port.device
        except serial.SerialException:
            pass
    return None
```

Fig. 2. Código em python utilizado para achar a porta, lendo as portas abertas, enviando na porta em que se lê a mensagem e envia a mensagem que foi achada

B. Enviando os parâmetros PID

O ESP32 aguarda os valores de PID e a velocidade inicial para iniciar a corrida. O computador solicita esses valores ao usuário e, após a validação dos quatro parâmetros, os transmite via Bluetooth para o ESP32. O computador aguarda a confirmação de recebimento dos dados pelo ESP32 antes de iniciar a contagem do tempo da corrida. Durante a corrida, o computador monitora a execução e aguarda o sinal de término.

C. Recebendo dados da ESP32

Ao recebê-lo, o tempo final é registrado e o usuário é solicitado a fornecer feedback sobre o desempenho do robô, incluindo se ele seguiu a linha com precisão e a oscilação na pista. Todos os dados coletados, incluindo o tempo da corrida, as notas do usuário e os parâmetros de controle, são armazenados em um banco de dados que foi desenvolvido utilizando o sistema de gerenciamento relacional SQLite para uma análise posterior. Estes registros serão utilizados em uma análise posterior. O resultado dessa análise, que poderá ser realizada com apoio de sistemas de aprendizagem de máquina, gerará novos valores de PID visando uma calibração mais precisa para os robôs.

III. APLICAÇÃO EM SALA PARA ALUNOS DA REDE PÚBLICA

Nos dias 14/05/2024 e 17/05/2024, foi oferecida uma formação para alunos do curso técnico de Desenvolvimento de Sistemas, utilizando ferramentas desenvolvidas durante o trabalho. As aulas foram ministradas no GARAGINO, laboratório de computação física da faculdade CESAR SCHOOL, e na ETE Porto Digital, uma escola técnica estadual escolhida por já possuir um grupo de estudos em robótica competitiva e oferecer uma disciplina eletiva de robótica.

Na primeira aula, realizada no GARAGINO para os alunos do grupo de robótica, o conceito de telemetria foi apresentado de forma expositiva, e os alunos participaram da execução prática de um projeto de pequeno porte e um de médio porte, com analogias aos sistemas de telemetria utilizados em tecnologias do dia a dia dos próprios estudantes.

Além disso, foi demonstrado o ambiente de desenvolvimento para a ESP32, utilizando o Arduino IDE, e foram fornecidas instruções para a instalação dos drivers necessários para programar a placa.

Os alunos também tiveram a oportunidade de montar um circuito eletrônico, seguindo um tutorial detalhado disponível no site Usinainfo (Usinainfo, 2024)[5]. Eles foram instruídos a utilizar a protoboard, fios jumpers e os componentes eletrônicos que foram fornecidos para montagem do sistema ilustrado na figura 3.

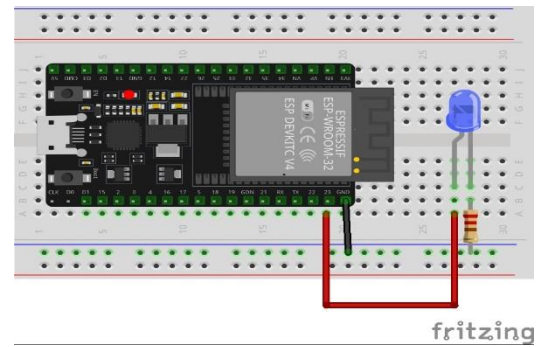


Fig. 3. Ilustração do modelo do sistema utilizado na aula

Durante a experimentação dos projetos, foram exploradas diferentes possibilidades de montagem. O conceito de telemetria e sua aplicação na IoT também foram discutidos. Foi fornecida uma explicação detalhada sobre o funcionamento do projeto, incluindo a lógica do código e o uso da biblioteca BluetoothSerial.h no sistema embarcado.

A. Comandos aprendidos

Para demonstrar aos alunos de modo simples como pode-se manusear a ESP32 usando comandos via cabo com a porta serial, foi criado um código onde o sistema ficava aguardando um comando do computador, enviado via porta serial como ilustrado na figura 4.

```
void setup() {
  Serial.begin(19200);
  pinMode(LED_PIN, OUTPUT);
  Serial.println("Digite 'ligar' para ligar o LED e 'desligar' para desligar o LED.");
}

void loop() {
  if (Serial.available()) {
    String command = Serial.readStringUntil('\n');
    command.trim();

    if (command.equals("ligar")) {
      digitalWrite(LED_PIN, HIGH);
      Serial.println("LED ligado");
    } else if (command.equals("desligar")) {
      digitalWrite(LED_PIN, LOW);
      Serial.println("LED desligado");
    } else {
      Serial.println("Comando desconhecido");
    }
  }
}
```

Fig. 4. Código utilizado para controle do LED com porta serial via cabo

Em seguida, utilizando o mesmo princípio, foi apresentado aos alunos como fazer a comunicação via porta serial com Bluetooth onde eles ligaram a placa o Bluetooth da placa e em seguida fizeram os comandos serem lidos da serial bBluetooth da placa, assim podendo ligar e desligar o LED sem precisar ter o computador ligado via cabo na placa como ilustrado na figura 5.

```
void setup() {
  Serial.begin(19200);
  SerialBT.begin("ESP32_BT"); // Nome do dispositivo Bluetooth
  pinMode(LED_PIN, OUTPUT);
  Serial.println("Digite 'ligar' para ligar o LED e 'desligar' para desligar o LED.");
}

void loop() {
  if (SerialBT.available()) {
    String command = SerialBT.readStringUntil('\n');
    command.trim();

    if (command.equals("ligar")) {
      digitalWrite(LED_PIN, HIGH);
      SerialBT.println("LED ligado");
    } else if (command.equals("desligar")) {
      digitalWrite(LED_PIN, LOW);
      SerialBT.println("LED desligado");
    } else {
      SerialBT.println("Comando desconhecido");
    }
  }
}
```

Fig. 5. Código utilizado para controle do LED com porta serial via Bluetooth

Ao receber o comando “ligar”, o sistema liga o LED e escreve na serial “LED ligado” e ao receber o comando “desligar”, o sistema desliga o LED e escreve na tela “desligado”, este código foi usado para exemplificar como os comandos na porta serial podem ser usados para manusear atuadores no sistema.

B. Aula em conjunto com eletiva de robotica

No segundo dia foi ministrado aula juntamente com a eletiva de robótica da escola, onde se abordou os a mesma base teórica aplicada no primeiro dia para os alunos da eletiva, porém com foco maior na base teórica onde foi abordado além dos assuntos do primeiro dia como funciona as portas seriais, lógica de programação para complementar na carga horária da eletiva, conceitos básicos de robótica e como funciona a robótica competitiva.

IV. CONCLUSÃO

A aplicação da telemetria para robôs autônomos pode ser feita usando o BLE com códigos simples e pode ser lido e enviado dados por qualquer dispositivo desde que conectado ao controlador via Bluetooth como o exemplificado neste artigo.

A utilização do Pyserial, na ótica deste trabalho pode ser utilizada como forma de facilitar a introdução dos conceitos de conexão, envio e recebimento de comandos entre robôs móveis e dispositivos de processamento.

Além disso, foi verificada a eficácia no ensino de tecnologias ligadas à robótica com a introdução dos conceitos e experimentação com alunos da rede pública estadual de ensino, desde a utilização de sistemas de telemetria até o desenvolvimento desses.

Sugere-se como estudos futuros a automação de testes de PID, com o acoplamento desses sistemas à pistas inteligentes de testes com detecção e comunicação, usando sistema de telemetria via Bluetooth, fazendo com que o sistema possa definir parâmetros otimizados de PID.

REFERENCIAS

- [1] K. Ogata, *Engenharia de Controle Moderna*, 5ª ed., São Paulo, Brasil: Pearson Prentice Hall, 2010, pp. 6–7.
- [2] K. Ogata, *Engenharia de Controle Moderna*, 5ª ed., São Paulo, Brasil: Pearson Prentice Hall, 2010, pp. 521.
- [3] Bluetooth SIG, "Bluetooth 5.2 Feature Overview," p. 6, Jan. 2020. [Online]. Available: https://www.bluetooth.com/wp-content/uploads/2020/01/Bluetooth_5.2_Feature_Overview.pdf.
- [4] M. Colom-Palacios, J. Cuervo-Gutiérrez, C. O'Mathuna, A. García-Castro, P. Popovski, and M. Calvo-Fernández, "Bluetooth Low Energy: A Survey", 2017.
- [5] UsinaInfo, "ESP32," [Online]. Available: <https://www.usinainfo.com.br/blog/esp32/>. [Accessed: Aug. 4, 2024].